POSTGRESQL - LIKE CLAUSE

The PostgreSQL **LIKE** operator is used to match text values against a pattern using wildcards. If the search expression can be matched to the pattern expression, the LIKE operator will return true, which is **1**.

There are two wildcards used in conjunction with the LIKE operator:

- The percent sign
- The underscore _

The percent sign represents zero, one, or multiple numbers or characters. The underscore represents a single number or character. These symbols can be used in combinations.

If either of these two signs is not used in conjunction with the LIKE clause, then the LIKE acts like the equals operator.

Syntax:

The basic syntax of % and _ is as follows:

```
SELECT FROM table_name
WHERE column LIKE 'XXXX%'

or

SELECT FROM table_name
WHERE column LIKE '%XXXX%'

or

SELECT FROM table_name
WHERE column LIKE 'XXXX_'

or

SELECT FROM table_name
WHERE column LIKE '_XXXX'

or

SELECT FROM table_name
WHERE column LIKE '_XXXX_'
```

You can combine N number of conditions using AND or OR operators. Here XXXX could be any numeric or string value.

Example:

Here are number of examples showing WHERE part having different LIKE clause with '%' and '_' operators:

Statement	Description
WHERE SALARY::text LIKE '200%'	Finds any values that start with 200
WHERE SALARY::text LIKE '%200%'	Finds any values that have 200 in any position
WHERE SALARY::text LIKE '_00%'	Finds any values that have 00 in the second and third positions

WHERE SALARY::text LIKE '2_%_%'	Finds any values that start with 2 and are at least 3 characters in length
WHERE SALARY::text LIKE '%2'	Finds any values that end with 2
WHERE SALARY::text LIKE '_2%3'	Finds any values that have a 2 in the second position and end with a 3
WHERE SALARY::text LIKE '2___3'	Finds any values in a five-digit number that start with 2 and end with 3

> *Postgres LIKE is String compare only. Hence, we need to explicitly cast the integer column to string as in the examples above.*

Let us take a real example, consider the table COMPANY having records as follows:

```
# select * from COMPANY;
 id | name  | age | address    | salary
----+-------+-----+-----------+--------
  1 | Paul  |  32 | California|  20000
  2 | Allen |  25 | Texas     |  15000
  3 | Teddy |  23 | Norway    |  20000
  4 | Mark  |  25 | Rich-Mond |  65000
  5 | David |  27 | Texas     |  85000
  6 | Kim   |  22 | South-Hall|  45000
  7 | James |  24 | Houston   |  10000
(7 rows)
```

Following is an example, which would display all the records from COMPANY table where AGE starts with 2:

```
testdb=# SELECT * FROM COMPANY WHERE AGE::text LIKE '2%';
```

This would produce the following result:

```
 id | name  | age | address     | salary
----+-------+-----+------------+--------
  2 | Allen |  25 | Texas      |  15000
  3 | Teddy |  23 | Norway     |  20000
  4 | Mark  |  25 | Rich-Mond  |  65000
  5 | David |  27 | Texas      |  85000
  6 | Kim   |  22 | South-Hall |  45000
  7 | James |  24 | Houston    |  10000
  8 | Paul  |  24 | Houston    |  20000
(7 rows)
```

Following is an example, which would display all the records from COMPANY table where ADDRESS will have a hyphen – inside the text:

```
testdb=# SELECT * FROM COMPANY WHERE ADDRESS   LIKE '%-%';
```

This would produce the following result:

```
 id | name | age |                address                | salary
----+------+-----+--------------------------------------+--------
  4 | Mark |  25 | Rich-Mond                            |  65000
  6 | Kim  |  22 | South-Hall                           |  45000
(2 rows)
```

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js